UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/697,420	10/30/2003	Michael P. Foley	MSFT-01153US0	7216

47766 7590 04/09/2008

VIERRA MAGEN/MICROSOFT CORPORATION
575 MARKET STREET, SUITE 2500
SAN FRANCISCO, CA 94105

EXAMINER	
HSU, JONI	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
04/09/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/697,420	FOLEY, MICHAEL P.	
	Examiner	Art Unit	
	JONI HSU	2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 January 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-6,8-11 and 13-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-6,8-11 and 13-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed January 16, 2008 are considered but are not persuasive.

2. Applicant argues Johns (US006421053B1) does not reorder data at all, but merely renders data as it is stored using unique serpentine method. That is, data is properly displayed, in its proper order—only method for rendering data to display has changed. Data has not been reordered before rendering. Zig-zag rendering is not reordering of data at all, but rather, just method for rendering that does not follow conventional scan line method (p. 8).

In reply, Johns teaches in order to perform zig-zag rendering, raster engine receives primitives for rendering, and breaks primitives into scan lines, and generates groups of scan lines to be rendered (c. 3, ll. 45-49). Groups of scan lines are transferred to blocking control logic (c. 3, ll. 58-60). Blocking control logic converts **(reorders)** each group of scan lines into 4x2 blocks of pixels (c. 4, ll. 5-7). So, Johns is considered to teach zig-zag rendering performs reordering, so reordering is performed using operation of secondary processor not provided for that function.

3. Applicant argues cited combination does not teach transforming original coordinates to new coordinates using operation of secondary processor not provided for that function (p. 9).

In reply, Examiner points out Childers (US005793996A) teaches reordering comprises transforming original coordinates to new coordinates (c. 17, ll. 54-c. 18, ll. 2). Data are subdivided into half-words (2 bytes) (c. 8, ll. 32-42), and so each subdivision is 2 bytes. Position of each subdivision is defined by byte address (c. 17, ll. 54-c. 18, ll. 2). Term "coordinate" is usually taken to mean magnitude that serves to define position of something. Since byte address serves to define position of subdivision, Childers teaches determining original positions of

coordinates defining each subdivision. Since Johns teaches reordering is performed using operation of secondary processor not provided for that function, as discussed above, and Childers expressly teaches reordering comprises transforming original coordinates to new coordinates, combination of Childers and Johns is considered to teach this limitation.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

6. Claims 21, 2-6, 13, 22, and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilson (US005960213A) in view of Childers (US005793996A), further in view of Johns (US006421053B1).

7. As per Claim 21, Wilson teaches method for reordering data from first predefined order to second predefined order in system having a primary processor (host processor) and secondary processor (GLINT Delta), comprising creating and storing data that are arranged in first predefined order (c. 2, ll. 47-49; c. 4, ll. 4-25). Wilson teaches generating byte swapped big-

endian data, commonly known as gib-endian. GLINT Delta accepts and converts gib-endian data (c. 4, ll. 20-24). Since process of converting gib-endian data includes byte swapping big-endian data, and byte swapping data is considered to be reordering data, this means that converting data is same as reordering data. According to Applicant's disclosure, reordering data using operation that was not provided for that purpose involves using texture operations to perform the reordering (p. 11, ll. 20-22). Wilson teaches GLINT Delta performs operations such as texture operations (c. 2, ll. 47-62), and since GLINT Delta is performing reordering, Wilson teaches reordering data using operation of secondary processor thereby obtaining reordered data that are arranged in second predefined order. Wilson teaches displaying data arranged in second predefined order (c. 4, ll. 20-25; c. 20, ll. 4-10; c. 3, ll. 60-62; c. 10, ll. 4-7).

However, Wilson does not teach determining subdivisions of data that are arranged in 1st predefined order, each subdivision is based on predefined size of each datum of data; determining original positions of coordinates defining each subdivision within data that are arranged in 1st predefined order; and causing 2nd processor to perform operation, which transforms coordinates of each subdivision to new positions and repositions data of each subdivision to have same locations relative to new positions as data had relative to original positions, thereby reordering data from first predefined order to second predefined order. But, Childers teaches each pixel contains 2 bytes (16 bits) of data (c. 4, ll. 61-64; Fig. 4A, 4B). Data are subdivided into half-words (16 bits) in order to ensure that pixels are presented in uniform format regardless of endian characteristic of bus from which they were received or of software or apparatus that generated pixels (c. 8, ll. 32-42), so pixels themselves do not become "scrambled" as result of their bytes being swapped (c. 4, ll. 48-c. 5, ll. 16). So, Childers teaches determining

subdivisions of the data that are arranged in first predefined order (little-endian, c. 20, ll. 40-45; c. 8, ll. 36-42), each subdivision is based on predefined size (2 bytes) of each datum (pixel) of data (c. 4, ll. 61-64; c. 8, ll. 32-42); determining original positions of coordinates defining each subdivision within data are arranged in first predefined order; and causing processor to perform operation, which transforms coordinates of each subdivision to new positions and repositions data of each subdivision to have same locations relative to new positions as data had relative to original positions, thereby reordering data from first predefined order to second predefined order (c. 17, ll. 54-c. 18, ll. 2). Applicant's disclosure describes coordinates defining each subdivision includes coordinates for the top, left, bottom, and right (p. 11, ll. 15-19). But, claims themselves do not define coordinates as including coordinates for top, left, bottom, and right. Childers teaches data are subdivided into half-words (2 bytes) (c. 8, ll. 32-42), and so each subdivision is 2 bytes. Position of each subdivision is defined by byte address (c. 17, ll. 54-c. 18, ll. 2). Term "coordinate" is usually taken to mean magnitude that serves to define position of something. Since byte address serves to define position of subdivision, Childers teaches determining original positions of coordinates defining each subdivision, as recited in claims. Childers teaches displaying data arranged in second predefined order (c. 17, ll. 54-c. 18, ll. 2; c. 7, ll. 45-49).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson to include determining subdivisions of data that are arranged in 1st predefined order, each subdivision is based on predefined size of each datum of data; determining original positions of coordinates defining each subdivision in data that are arranged in 1st predefined order; causing secondary processor to perform operation, which transforms coordinates of each subdivision to new positions and repositions data of each subdivision to have

same locations relative to new positions as data had relative to original positions, so reordering data from 1st predefined order to 2nd predefined order because Childers teaches guaranteeing pixels are placed into their proper location and bytes in pixel are in proper order (c. 4, ll. 48-c. 5, ll. 16; c. 8, ll. 32-42; c. 17, ll. 54-c. 18, ll. 2).

But, Wilson and Childers do not explicitly teach reordering is performed by using operation intended for another function. But, Johns describes in prior art, pixels are accessed in fixed groups usually organized as a span in the horizontal or vertical direction (c. 1, ll. 32-38). Johns invention is directed to being able to render block-by-block in serpentine manner from entry block, first in direction away from long edge of primitive and then in direction towards long edge. Blocks in different span subgroups are then alternately rendered, such that rendering zig-zags between span subgroups as it proceeds to end of span group. Once first end of span group is reached, rendering resumes from entry block in opposite direction, but in same zig-zag manner, until other end of span group is reached (c. 2, ll. 4-25). So, Johns teaches rendering operation is able to reorder data in span so span can be rendered in a zig-zag manner. So, Johns teaches reordering is performed by using operation intended for another function (rendering). In order to perform zig-zag rendering, raster engine receives primitives for rendering, and breaks primitives into scan lines, and generates groups of scan lines to be rendered (c. 3, ll. 45-49). Groups of scan lines are transferred to blocking control logic (c. 3, ll. 58-60). Blocking control logic converts (**reorders)** each group of scan lines into 4x2 blocks of pixels (c. 4, ll. 5-7). So, Johns is considered to teach zig-zag rendering performs reordering.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson and Childers so reordering is performed by using an operation

intended for another function because Johns teaches rendering can be optimized for both small scan lines and texture mapping, and memory bandwidth use between pixel/texel cache and frame buffer is improved to reduce pixel/texel fetches required for rendering (c. 1, ll. 32-c. 2, ll. 28).

8. As per Claim 2, Wilson teaches secondary processor has graphics processor (c. 2, ll. 43-48, 63-64; c. 4, ll. 20-25; c. 20, ll. 4-10).

9. As per Claim 3, Wilson does not expressly teach what 1st and 2nd predefined orders are. According to Applicant's disclosure, pixilated endian order means data is subdivided as function of pixel size (p. 4, ll. 10-12). So, Childers teaches one of following conditions exists: (c) 1st predefined order has pixilated little endian order, 2nd predefined order has big endian order; (d) 1st predefined order has pixilated big endian order, 2nd predefined order has little endian order (c. 8, ll. 36-42; c. 20, ll. 40-45).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson so one of following conditions exists: (c) 1st predefined order has pixilated little endian order, 2nd predefined order has big endian order; (d) 1st predefined order has pixilated big endian order, 2nd predefined order has little endian order because Childers teaches interconnecting 2 buses of otherwise incompatible types (c. 5, ll. 31-33) and guaranteeing pixels are placed into their proper location and also bytes in pixel are in proper order (c. 4, ll. 48-c. 5, ll. 16; c. 8, ll. 32-42; c. 17, ll. 54-c. 18, ll. 2).

10. As per Claim 4, Wilson teaches data has image data (c. 2, ll. 43-48, 63-64; c. 4, ll. 20-25; c. 20, ll. 4-10).

11. As per Claim 5, Wilson doesn't using draw operation or multi-texture draw operation to perform reordering. However, Johns teaches this limitation, as discussed for Claim 21.

12. As per Claim 6, Wilson does not teach creating and storing step has steps of defining secondary storage space that is accessible to secondary processor; primary processor storing data in 1st predefined order in primary storage space that is accessible to both primary processor and secondary processor; and secondary processor copying data in 1st predefined order from primary storage space to secondary storage space. But, Childers teaches creating and storing step has steps of defining secondary storage space 523 that is accessible to secondary processor; primary processor 531 storing data in 1st predefined order in primary storage space 517 that is accessible to both primary processor and secondary processor; and secondary processor copying data in 1st predefined order from primary storage space to secondary storage space (c. 8, ll. 3-6; c. 8, ll. 29-32; c. 10, ll. 61-c. 11, ll. 1; c. 6, ll. 61-64; c. 20, ll. 41-47; c. 7, ll. 49-58).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson so creating and storing includes defining secondary storage space that is accessible to secondary processor; primary processor storing data in 1st predefined order in primary storage space that is accessible to both primary processor and secondary processor; and secondary processor copying data in 1st predefined order from primary storage space to secondary storage space because Childers teaches being able to write intelligible pixels into frame buffer from source on either of two endian-incompatible buses (c. 5, ll. 37-39).

13. As per Claim 13, Wilson teaches fetching commands using DMA controller (c. 4, ll. 7-10), and so commands are inherently fetched from memory medium, and so there is memory medium on which are stored machine instructions for carrying out steps.

14. As per Claim 22, it is similar in scope to Claim 21, except Claim 22 is for transforming using draw operation. Johns teaches this, as discussed for Claim 21. So, Claim 22 is rejected under same rationale as Claim 21.

15. As per Claim 23, Wilson does not teach draw operation used for transforming comprises multi-textured draw operation. However, Johns teaches this limitation (c. 1, ll. 63-c. 2, ll. 25).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson so draw operation used for transforming has multi-textured draw operation as taught by Johns. Johns teaches multi-textured draw operations make image more realistic looking. Using this for transforming is advantageous because texture mapping requires many texel fetches, by enabling multi-textured draw operation to transform data, this enables texel reuse during texture mapping to reduce total number of texel fetches, which improves memory bandwidth use between texel cache and frame buffer (c. 1, ll. 32-53; c. 2, ll. 21-25).

16. Claims 8-11 and 14-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilson (US005960213A), Childers (US005793996A), and Johns (US006421053B1) in view of Baldwin (US005594854A).

17. As per Claim 8, Wilson and Childers are relied upon for teachings relative to Claim 1.

However, Wilson and Childers do not teach determining original coordinates of each subdivision includes determining vertices of each subdivision relative to origin of data. However, Baldwin teaches reordering data from big-endian order to little-endian order (c. 23, ll. 20-29) and preserving order of pixels as well as byte ordering within each pixel (c. 23, ll. 42-49). Data is subdivided into blocks or triangles (c. 38, ll. 52-53; c. 31, ll. 14-15; c. 35, ll. 53-64). Baldwin

teaches determining original coordinates of each subdivision includes determining vertices of each subdivision relative to origin of data (c. 32, ll. 41-49).

It would be obvious to one of ordinary skill in the art at time of invention by applicant to modify Wilson-Childers to include determining original coordinates of subdivisions including determining vertices of subdivisions relative to origin of data because Baldwin teaches being able to perform simple add operations so no new arithmetic elements are needed (c. 5, ll. 26-61).

18. As per Claim 9, Wilson does not teach transforming the original coordinates includes instructing secondary processor to transpose coordinates of each subdivision so as to mirror data of each subdivision about center position. However, Baldwin teaches this (c. 39, ll. 35-45).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson so transforming original coordinates secondary processor transposing coordinates subdivisions so as to mirror data of each subdivision about center position because Baldwin suggests being able to reorder data without complex processing (c. 39, ll. 35-45).

19. As per Claim 10, Wilson doesn't teach predefining mask for selectively retaining subset of data; applying mask to subdivisions to subdivide data into subsets of data that are iteratively repositioned to new locations relative to new coordinates, new locations for original locations relative to original coordinates. But, Baldwin teaches this limitation (c. 51, ll. 11-22).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson to include predefining mask for selectively retaining subset of data; applying mask to subdivisions to subdivide data into subsets that are iteratively repositioned to new locations relative to new coordinates, new locations corresponding to original locations

relative to original coordinates because Baldwin suggests this defines limits of block to be written, thereby reducing amount of processing that must be performed (c. 51, ll. 11-22).

20. As per Claim 11, Wilson does not teach determining portion of data that changed since previous execution cycle so only that portion is reordered between 1st and 2nd predefined orders. But, Baldwin teaches when begin-draw command is sent, internal registers are updated, but if continue-draw command is sent, then this update does not happen and drawing continues with current values in internal registers (c. 13, ll. 27-31). So, reordering only occurs when data is changed. So, Baldwin inherently teaches determining portion of data that changed since previous execution cycle so only portion of data that changed since previous execution cycle is reordered between 1st and 2nd predefined orders (c. 13, ll. 27-31; c. 23, ll. 20-29, 37-51).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson to include determining portion of data that changed since previous execution cycle so only that portion is reordered between 1st and 2nd predefined orders because Baldwin suggests eliminating unnecessary processing (c. 31, ll. 27-31).

21. As per Claim 14, Wilson teaches generating byte swapped big-endian data (gib-endian). GLINT Delta accepts and converts gib-endian data (c. 4, ll. 20-24). Since converting gib-endian data includes byte swapping big-endian data, and byte swapping data is considered to be reordering data, this means that converting data is same as reordering data. According to Applicant's disclosure, reordering data using operation that was not provided for that purpose involves using texture operations to perform reordering (p. 11, ll. 20-22). Wilson teaches GLINT Delta performs operations such as texture operations (c. 2, ll. 47-62), and GLINT Delta (secondary processor) reorders data between 1st predefined order and 2nd predefined order (c. 2,

ll. 43-64; c. 4, ll. 20-25; c. 20, ll. 4-10). Wilson teaches primary processor (host processor); secondary processor (GLINT Delta) in communication with primary processor (c. 3, ll. 60-62); memory in communication with primary processor and secondary processor, memory storing data in 1st predefined order (c. 3, ll. 60-c. 4, ll. 25). Commands are fetched using DMA controller (c. 4, ll. 7-10), so commands are fetched from memory, so machine instructions are stored that cause primary processor and secondary processor to carry out functions, including creating and storing data that are arranged in 1st predefined order (c. 2, ll. 43-64; c. 4, ll. 4-25; c. 20, ll. 4-10).

However, Wilson does not teach determining subdivisions of data that are arranged in 1st predefined order, wherein each subdivision is based on predefined size of each datum of data; determining original coordinates defining each subdivision within data that are arranged in 1st predefined order; and using secondary processor for performing operation, thereby transforming original coordinates of each subdivision to new coordinates and repositioning data of each subdivision to have same locations relative to new coordinates as data had relative to original coordinates, thereby reordering data from 1st predefined order to 2nd predefined order. However, Childers teaches each pixel contains 2 bytes (16 bits) of data (c. 4, ll. 61-64; Fig. 4A, 4B). Data are subdivided into half-words (16 bits) in order to ensure that pixels are presented in uniform format regardless of endian characteristic of bus from which they were received or of software or apparatus that generated pixels (c. 8, ll. 32-42), so pixels themselves do not become "scrambled" as result of their bytes being swapped (c. 4, ll. 48-c. 5, ll. 16). So, Childers teaches determining subdivisions of data that are arranged in 1st predefined order (little-endian, c. 20, ll. 40-45; c. 8, ll. 36-42), each subdivision is based on predefined size (2 bytes) of each datum (pixel) of data (c. 4, ll. 61-64; c. 8, ll. 32-42); determining original coordinates defining each subdivision within

data that are arranged in 1st predefined order; and using secondary processor for performing operation, so transforming original coordinates of each subdivision to new coordinates and repositioning data of each subdivision to have same locations relative to new coordinates as data had relative to original coordinates, so reordering data from 1st to 2nd predefined order (c. 17, ll. 54-c. 18, ll. 2). Applicant's disclosure describes coordinates defining each subdivision includes coordinates for the top, left, bottom, and right (p. 11, ll. 15-19). But, claims themselves do not define coordinates as including coordinates for the top, left, bottom, and right. Childers teaches data are subdivided into half-words (2 bytes) (c. 8, ll. 32-42), and so each subdivision is 2 bytes. Position of each subdivision is defined by byte address (c. 17, ll. 54-c. 18, ll. 2). Term "coordinate" is taken to mean magnitude that serves to define position of something. Since byte address serves to define position of subdivision, Childers teaches determining original positions of coordinates defining each subdivision. This would be obvious for reasons for Claim 21.

But, Wilson, Childers do not expressly teach reordering by using operation of secondary processor provided for different purpose. But, Johns teaches this, as discussed for Claim 21.

However, Wilson, Childers, and Johns do not teach primary processor determines subdivisions of data; using primary processor, determining original positions of subdivisions within data. However, Baldwin teaches framebuffer bypass determines subdivisions of data; using framebuffer bypass, determining original positions of subdivisions within data (c. 23, ll. 20-29, 37-51; c. 32, ll. 41-49). Host processor controls framebuffer bypass to access frame buffer (c. 21, ll. 28-46), and so primary processor is doing this.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson, Childers, Johns so primary processor determines subdivisions of

data; primary processor determines original positions of subdivisions in data because Baldwin teaches primary processor directly accesses frame buffer for faster processing (c. 21, ll. 28-46).

22. As per Claim 15, it is similar in scope to Claim 3, and so is rejected under same rationale.

23. As per Claim 16, Wilson does not teach instructions cause processor to determine size of each subdivision as function of predefined size of each datum of data; and determining number of subdivisions within data. But, Childers teaches machine instructions cause processor to determine size of each subdivision as function of predefined size (2 bytes) of each datum (pixel) of data (c. 4, ll. 61-64; Fig. 4A, 4B; c. 8, ll. 32-42); and determining number of subdivisions (4) within data (c. 9, ll. 31-38). This would be obvious for reasons given for Claim 1.

But, Wilson and Childers do not teach primary processor determines subdivision. But, Baldwin teaches this (c. 23, ll. 36-51; c. 21, ll. 28-46). This is obvious for reasons for Claim 14.

24. As per Claim 17, Wilson does not teach machine instructions cause primary processor to determine vertices of each subdivision relative to origin of data. However, Baldwin teaches reordering data from big-endian order to little-endian order (c. 23, ll. 20-29) and preserving order of pixels as well as byte ordering within each pixel (c. 23, ll. 42-49). Data is subdivided into blocks or triangles (c. 38, ll. 52-53; c. 31, ll. 14-15; c. 35, ll. 53-64). Machine instructions cause primary processor to perform function of determining vertices of each subdivision relative to origin of data (c. 13, ll. 16-17, 53-55). This would be obvious for reasons given for Claim 8.

25. As per Claim 18, it is similar in scope to Claim 9, and so is rejected under same rationale.

26. As per Claim 19, Wilson does not teach machine instructions cause primary processor to perform function of predefining mask for selectively retaining data subset and cause secondary processor to perform function of applying mask to subdivisions to subdivide data into plurality of

subsets of data that are iteratively repositioned to new locations relative to new positions, new locations corresponding to original locations relative to original positions. But, Baldwin teaches this (c. 51, ll. 11-22; c. 53, ll. 39-40; c. 13, ll. 16-17). This is obvious for reasons for Claim 10.

27. As per Claim 20, Wilson does not expressly teach display in communication with secondary processor, machine instructions cause secondary processor to perform function of displaying data arranged in 2nd predefined order on display. But, Childers teaches display 515 in communication with secondary processor 511, instructions cause secondary processor to perform function of displaying data arranged in 2nd predefined order on display (c. 8, ll. 36-43).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Wilson to include display in communication with secondary processor, instructions cause secondary processor to display data arranged in second predefined order on display because Childers teaches if video input device is connected to bus that is incompatible with bus connected to display, data must be reordered in order to be displayed (c. 3, ll. 15-25).

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of advisory action. In no event, however, will statutory period for reply expire later than SIX MONTHS from date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JONI HSU whose telephone number is (571)272-7785. The examiner can normally be reached on M-F 8am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kee Tung can be reached on 571-272-7794. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JH

/Kee M Tung/
Supervisory Patent Examiner, Art Unit 2628